Report No. UIUCDCS-R-76-823 NSF-OCA-DCR73-07980 A02-000022

EVALUATION OF SELECTED FEATURES OF THE EUREKA FULL-TEXT INFORMATION RETRIEVAL SYSTEM

by

James Richard Rinewalt

September 1976

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

Report No. UIUCDCS-R-76-823

EVALUATION OF SELECTED FEATURES OF THE EUREKA FULL-TEXT INFORMATION RETRIEVAL SYSTEM*

by

James Richard Rinewalt

September 1976

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

*This work was supported in part by the National Science Foundation under Grant No. US NSF-DCR73-07980 A02 and was submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science, September 1976.

EVALUATION OF SELECTED FEATURES OF THE EUREKA FULL-TEXT INFORMATION RETIREVAL SYSTEM

James Richard Rinewalt, Ph. D.
Department of Computer Science
University of Illinois at Urbana-Champaign, 1976

Evaluations of on-line information retrieval systems have been largely dependent upon monitoring of users' searches and the cooperation of users in interviews and questionaires. Since users have a variety of information needs and levels of experience, the evaluation process has been difficult.

This paper describes and presents the results of a series of experiments designed to evaluate various features of an information retrieval system in a controlled environment. Features are evaluated on the basis of their development and implementation cost, their effect on system performance as well as user performance, and the attitude of users toward the feature.

ACKNOWLEDGMENT

The author would like to thank the Illinois Legislative Information Systems and West Publishing Company for making available a digitized copy of the Illinois State Statutes. He would also like to thank Bernie Hurley for his recruiting activities and Keith Morgan for helping run the classes. Also, the financial assistance of the Department of Computer Science and LTV Aerospace Corp. is gratefully acknowledged.

He would especially like to thank his advisor, Prof. Dave Kuck, for his advice, encouragement, and patience.

Finally, the author wishes to thank his parents, his wife Ruth Ann, and son Mark for their support and understanding.

TABLE OF CONTENTS

CHAPTER 1 — INTRODUCTION

In the 19 years since the start of the Cranfield Project, much work has been done on the problem of evaluating information retrieval systems. The early systems were of the controlled vocabulary type which operated in the batch mode and whose response to a search was a list of bibliographic citations. The primary tradeoff in this type of system was the cost of the depth and exhaustivity of indexing versus their effect on recall and precision. The primary concern of the user was to construct one complex search which would satisfy his need. Thus, the evaluation of this type of system was based on the values of its recall ratio (the proportion of relevant material actually retrieved in response to a search request) and its precision ratio (the proportion of retrieved material which is actually relevant). Cleverdon[1] listed other criteria but considered these two the most important. Cooper[2] proposed a measure of retrieval effectiveness which combined recall and precision and took into account the amount of relevant material desired by the user. Various other measures, all involving some form of precision and recall, have been proposed[3].

Advances in computer systems have made on-line full-text information retrieval systems practical. Since users are now able to conduct iterative searches, other criteria including user effort, response time, and the form in which search results are displayed have become more important[4].

The evaluation of operational on-line systems[4] has been largely dependent upon monitoring of users' searches and the cooperation of users in interviews and questionaires. Since the users had a variety of information needs and levels of experience, the evaluation process was quite difficult.

This thesis describes and presents the results of a series of experiments designed to evaluate various features of an information retrieval system in a controlled environment by comparing the performance of several users who have a common information need and somewhat comparable backgrounds. Carlisle[5] proposed a framework (Figure 1.1) for conducting research in man-computer interactions. In this framework, the system refers to items which are transparent to the user (e.g., the hardware, the language in which the routines are written, etc.) while the user-system interface refers to items directly affecting the user (e.g., the commands available, the command syntax, the form of the output, etc.). This thesis will examine the effect of varying two entities of man-computer interactions, the user-system interface and the task, on some of the characteristics of performance. Varying the user-system interface will consist of denying selected features to different groups of users, while the two tasks to be considered are short answer-type quizzes and essay-type quizzes. The characteristics of performance are highly interdependent. Those to be examined in this thesis are time, cost, and quantity and quality of performance.

In Chapter 2, the effects of inverted file structure on full-text retrieval systems will be discussed. A brief description of the EUREKA query language will be presented in Chapter 3. In Chapter 4, the design of

the experiments will be discussed and the results presented. The results of a survey of user attitudes will be discussed in Chapter 5. Chapter 6 is devoted to a summary of the results and some suggestions for future experiments.

Entities of Man-Computer Interation

1. The System
2. The Data Base
3. The User-System Interface
4. The User
5. The Training
6. The Setting
7. The Task

Characteristics of Performance

1. The time to perform the task
2. The cost to perform the task
3. The quantity and quality of the performance
4. The errors committed
5. The user's satisfaction
6. The utilization of available resources
7. The patterns of user and system behavior

Figure 1.1 Experimental Framework for Man-Computer Interaction Research[5]

CHAPTER 2 — INVERTED FILE STRUCTURE FOR FULL-TEXT IR SYSTEMS

For the sake of efficiency and adequate response time, an on-line full-text information retrieval system requires some form of inverted file or index to the words used in the text. Without this, the full text of each document, or some surrogate thereof, would have to be searched for each query submitted to the system. While this technique is straightforward, it is obviously time consuming.

The content of the inverted file varies from one implementation to another. For the purposes of this discussion, the inverted file structure of EUREKA[6] will be assumed:

1. A token is defined as any unbroken string of alphanumeric characters.
2. A type is a distinct token.
3. The inverted file contains only those types which occur in the data base.
4. Associated with each type is a list of pointers indicating where the tokens of this type occur in the data base.

The level to which the inverted file points is important to the design of a full-text information retrieval system. The level of indexing, in order of increasing specificity, may be that of document, section of a document, paragraph, sentence, or word. There may be other levels

appropriate for specific data bases. The tradeoff is the use of a higher, less specific level of indexing to conserve storage space versus a lower, more specific level of indexing to minimize full-text searching and improve response time.

Figure 2.1 illustrates the difference in storage requirements for various levels of indexing for a data base consisting of a set of state statutes. The area under a curve is the number of pointers required in the inverted file. Since the curves are plotted on log-log paper, the relative sizes of the areas may be misleading. Also, as the level of indexing becomes more specific, the number of bits required for each pointer increases. Table 2.1 shows the number of pointers required and the storage space used as a percentage of the full text for four levels of indexing for this data base.

INDEXING LEVEL	POINTERS REQUIRED	STORAGE USED (% FULL TEXT)
Document	0.39×10^6	7
Section	0.96×10^6	20
Paragraph	2.04×10^6	75
Word	3.30×10^6	120

Table 2.1 Storage Requirements for Different Indexing Levels

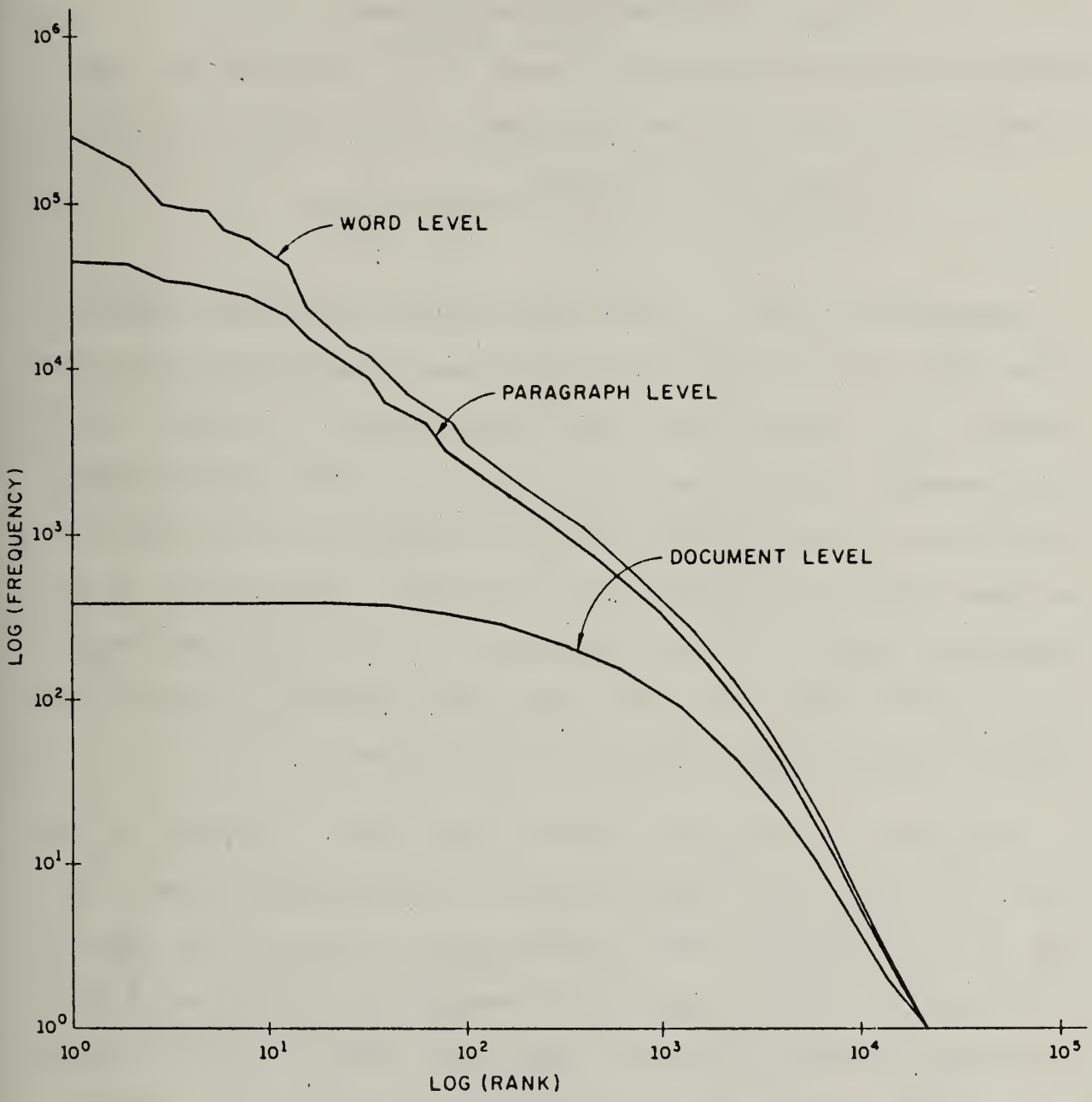


Figure 2.1 - Zipf Curves for State Statutes

Theoretically, the minimum number of bits required for indexing to the word level is given by

$$n \log_2 n \qquad \text{where n=number of tokens}$$

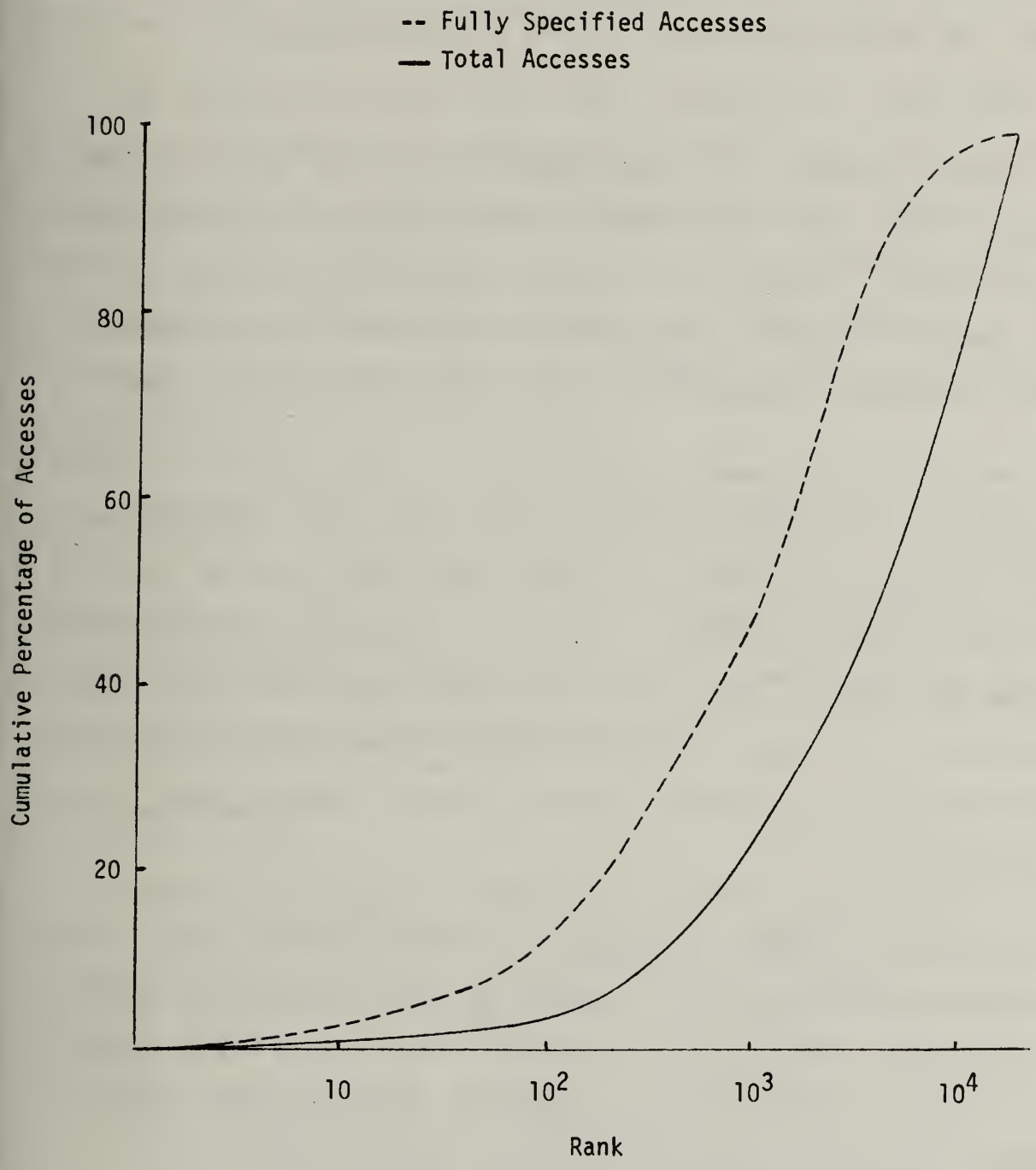
Assuming eight bit characters and an average of six characters per token, the size of the inverted file as a percentage of the full text is then

$$\frac{\text{inverted file}}{\text{full text}} = \frac{n \log_2 n}{48n} \times 100 \approx 2 \log_2 n$$

Realistically, however, pointer length should be an integral number of bytes. Also, pointers should distinguish between documents and subdivisions thereof; i.e., a pointer should indicate the document, the section within that document, the paragraph within that section, etc. This arrangement allows a user to search for co-occurrences of tokens at any level and allows the data base to change without total reinversion. Unfortunately, it also drastically increases the storage requirements; for the state statutes data base (3.3×10^6 tokens and 21202 types), the inverted file requires 120% instead of the theoretical minimum 43% of the full text.

Intuitively, not all types in the data base need be included in the inverted file. The high frequency types are generally syntactic words (THE, AND, OF, etc.), while the lowest frequency types are generally too specific to be useful in searches. One method of determining the usefulness of types is to count the number of times each was actually used in a search. Types can be accessed in two ways - (1) fully-specified or (2) truncated to eliminate prefixes and/or suffixes. Figure 2.2 shows data which was collected for the state statutes data base during the user experiments. The data is based on approximately 12000 searches conducted during 870

Figure 2.2 Token Useage for Statutes Data Base



user-hours. There were 10317 fully-specified and 31784 truncated types accessed during this period. Figure 2.2 indicates that 25% of the types (ranks 100 through 5192) account for 80% of the fully-specified types used for searching.

For this data base, the type-token ratio is 150 and the highest frequency token, THE, occurs 256×10^3 times. By deleting the sixty highest frequency tokens, the size of the inverted file containing pointers to the word level can be halved. Consequently, the type-token ratio is reduced to 75 and the highest frequency token of those remaining occurs 6116 times.

The choice of the level of indexing has a direct effect on system performance. If a high level is chosen and a large number of search requests specify a lower level, the system will spend much of its time performing full-text searching. This will increase response time and decrease the number of users which the system can effectively handle. Alternatively, if a lower level of indexing is chosen, storage will be used inefficiently unless a sufficient number of search requests occur at or below that level.

Among other features, this thesis will explore document-level versus section-level indexing. In an attempt to compare section-level indexing with paragraph-, sentence-, or word-level indexing, section-level indexing with full-text searching will be compared to that without full-text searching.

CHAPTER 3 — EUREKA QUERY LANGUAGE

The EUREKA query language has been designed as a basic tool for studying user behaviour and searching techniques. Most of the facilities provided in this system are available elsewhere, though not necessarily all together in one system. This chapter will give a brief description of the EUREKA language emphasizing those features which were evaluated in this research. A complete description of the language and a detailed explanation of its implementation are given in [6].

Each EUREKA user is given a file, accessible only by him, on disk in which a record of his actions is kept. The text of each query is stored here along with a list of identifiers for all documents responding to the query. This file also contains the text of any comments which the user has attached to a query or to individual documents.

3.1 Query Language

Currently, there are nine commands in the EUREKA query language. Only two of these (FIND and PRINT) are necessary for conducting searches, while the other seven perform auxiliary functions. In brief, the functions of the commands are:

FIND:

> The FIND statement is used to perform searches for documents containing a user selected set of words, parts of words, or phrases. The collection of document

identifiers returned by the FIND statement is known as a query set. This command will be discussed in more detail in section 3.1.1.

PRINT:

The PRINT command is used to print user comments, selected portions of a document, and information about preceding queries and their resultant query sets. This command will be discussed further in section 3.1.2.

MAKE:

The MAKE statement is used to compare and combine sets of documents created by previous FIND and MAKE statements.

COMMENT:

The COMMENT statement is used to write notes in the user file concerning a query set or particular document. These notes may be retrieved at a later time by use of the PRINT statement or, if attached to a document, searched by a FIND statement.

MACRO:

The MACRO statement is used to name lists of search terms so that the user does not have to repeatedly type in long search expressions. These macro definitions are saved in the user file and may be used in conjunction with other search terms in FIND statements.

CHANGE:

The CHANGE statement is used to assign a name to or change the existing name of a query set.

DELETE:

The DELETE statement is used to delete query sets and/or comments which are no longer needed.

LOGON:

The LOGON statement is used to identify the user to EUREKA in order for EUREKA to gain access to the correct user files and data base.

LOGOFF

The LOGOFF command is used to terminate a session. It disconnects a user from EUREKA and closes his files.

Each of these commands has a simple basic form. The FIND and PRINT commands, however, have optional clauses and modes of operation that significantly increase their power. These two commands will be discussed in more detail in the following paragraphs.

3.1.1 FIND Statement

The general form of the FIND statement is

```
FIND <search expression> [IN <context>]
     [FROM <set expression>]
     [= <set name>] ["<comments>"]
```

where <search expression> is an arbitrarily complex Boolean expression whose variables are search terms and whose operators are + and * representing the Boolean OR and AND operations, respectively. Search terms are enclosed in apostrophes and may consist of words, parts of words, phrases, or arbitrary character strings. A universal character, #, is provided for indicating to the system that prefixes, suffixes, or both have been deleted from a search term.

The clauses enclosed in square brackets are optional. The IN clause restricts the search to specified contexts of a document; e.g., author list, title, abstract, body, footnotes, etc. If this clause is omitted, the entire document is searched. The FROM clause can be used to restrict the search to specific documents or to results of previous queries. <set expression> is a Boolean expression whose variables are sets of documents and whose operators are +, *, and - representing the OR, AND and AND NOT operations. The last two optional clauses allow the user to assign an alphanumeric name to the query and to attach an arbitrary character string as a comment.

EUREKA's inverted file contains all types which occur in the data base. Associated with each type is a list of documents in which the type occurs. For each document in this list, there is a set of flags indicating which contexts of the document contain one or more occurrences of the type. Thus, many search requests can be satisfied by a search of this file. A search of the full text of a document, however, is required whenever the user (1) enters a search term containing nonalphanumeric characters, (2) searches for a co-occurrence of two or more terms in the same paragraph or sentence, or (3) searches his comments. Statistics, gathered during the user experiments, concerning the use of full-text searching will be presented in Chapter 4.

3.1.2 PRINT Statement

The PRINT statement has three uses. It may be used to display all or selected contexts of any document. It may also be used to display information about previous queries and the documents which responded to them

and to display macro definitions. Only the first use will be discussed here. The general form of the PRINT command for displaying a document or a selected part thereof is

PRINT <context list> FROM (<set ID> | [<document list>])

where <context list> specifies one or more contexts and the FROM clause indicates which documents are to be displayed. The argument of the FROM clause may be the user-assigned name or the system-assigned number of a set of documents created by a previous query, or a list of document accession numbers enclosed in square brackets.

The documents are displayed in order of probable relevance according to the frequency of occurrence of the search terms. When the system displays a portion of a document, the user has the option of browsing through the text. Using the currently displayed portion as an entry point, the user may move backward or forward one or more paragraphs or sentences, or he may display any other context of the document. He may at any time stop browsing and continue with the current display, skip to the next document in the output list, or cancel all further output. Also at any time he may attach a comment to the document currently being displayed by entering an arbitrary character string enclosed in quotes. Comments attached to a document may be displayed by a print statement or searched by a FIND statement.

3.2 Sample User Session

To illustrate the use of some of these commands, a sample user session is shown in Figure 3.1. All input by the user is underlined. The lines following "... DOCUMENTS ARE POSTED..." in each FIND statement give a list

of the accession numbers of the documents which responded to the search. The numbers in parentheses in these lines are the system-assigned relevance ranks based on the frequency of occurrence of the search terms.

The data base being used is a set of state statutes and the object of the session is to find the penalty for robbery. With a less restricted data base, the user might well be satisfied with the results of the first search, having retrieved only thirteen documents, and immediately begin viewing text. Indeed, the ranking mechanism would have presented him with the desired information in the first document it displayed. However, to illustrate more features of the language and since it is known that the desired information occurs in only one document, the object of the session is to retrieve and display only that document.

MAN-MACHINE INTERACTION	EXPLANATION
EUREKA VERSION 3.01 23-NOV-75	
PLEASE LOGON #LOGON DICK	The user identifies himself.
*** COMMAND COMPLETE	
QUERY #00001 #FIND 'ROBBERY'	He requests all documents containing the word ROBBERY.
13 DOCUMENTS REPONDED TO INVERTED FILE SEARCH 0 DOCUMENTS REQUIRED FULL-TEXT SEARCH 13 DOCUMENTS ARE POSTED TO THIS SET 64(6) 139(8) 160(13) 174(4) 188(5) 191(3) 197(7) 269(2) 271(9) 273(10) 295(11) 318(12) 357(1) *** COMMAND COMPLETE	
QUERY #00002 #FIND 'PENALTY' FROM LAST	He attempts to narrow the request by searching the documents which responded to the last query for the word PENALTY.
9 DOCUMENTS RESPONDED TO INVERTED FILE SEARCH 0 DOCUMENTS REQUIRED FULL-TEXT SEARCH 9 DOCUMENTS ARE POSTED TO THIS SET 64(1) 139(5) 160(4) 174(3) 188(2) 191(6) 197(9) 295(7) 318(8) *** COMAND COMPLETE	
QUERY #00003 #FIND 'ROBBERY'*'PENALTY' IN PARAGRAPH	Now he tries to find documents which contain both words in the same paragraph.
9 DOCUMENTS RESPONDED TO INVERTED FILE SEARCH 9 DOCUMENTS REQUIRED FULL-TEXT SEARCH 0 DOCUMENTS ARE POSTED TO THIS SET *** COMMAND COMPLETE	

Figure 3.1 Sample User Session on EUREKA

MAN-MACHINE INTERACTION	EXPLANATION
QUERY #00004 #MACRO 'PENAL#'+'SENTENCE#'+'#PRISON#' +'FINE#'=PENALTY	He creates a Macro containing some synonyms for PENALTY.
QUERY #00004 #FIND 'ROBBERY'*PENALTY IN PARAGRAPH 10 DOCUMENTS RESPONDED TO INVERTED FILE SEARCH 10 DOCUMENTS REQUIRED FULL-TEXT SEARCH 1 DOCUMENTS ARE POSTED TO THIS SET 357(1) *** COMMAND COMPLETE	He searches for documents containing the word ROBBERY and any one of his synonyms for PENALTY together in the same paragraph.
QUERY #00005 #PRINT PARAGRAPH FROM LAST * (B) SENTENCE. ROBBERY IS A CLASS 2 FELONY. !-PARAGRAPH * 18-1. ROBBERY.) (A) A PERSON COMMITS ROBBERY WHEN HE TAKES PROPERTY FROM THE PERSON OR PRESENCE OF ANOTHER BY THE USE OF FORCE OR BY THREATENING THE IMMINENT USE OF FORCE. !"DEFINITION & PENALTY FOR ROBBERY" !END *** COMMAND COMPLETE	He then asks the system to display the paragraphs containing those search terms. He enters Browse Mode by moving backward one paragraph to see if there is any more pertinent information. He attaches a comment to this document for possible later use. He exits from Browse Mode.
QUERY #00006 #LOGOFF *** COMMAND COMPLETE	Having found the required information, he terminates the session.

Figure 3.1 continued

CHAPTER 4 -- USER EXPERIMENTS

The experiments centered around machine assignments in a special topics course in information retrieval. The system used was a minicomputer-based experimental retrieval program known as EUREKA[6]. Two data bases were used: (1) a collection of thirty-seven technical articles on information retrieval containing approximately one million characters and (2) a set of state statutes containing approximately twenty million characters. Due to its small size, the information retrieval data base was used only during the first series of experiments.

4.1 Initial User Experiments

The initial set of experiments was intended to "shake down" the system and obtain a preliminary view of user reactions to it. The first set of experiments was conducted during the spring semester of 1975. Before registration and again at the first class meeting, the nature of the course was explained and students who wished to withdraw were given a chance to do so. The group which completed the course consisted of five graduate students and seven undergraduates. Four were majors in Computer Science, one in Engineering, one in Library Science, and six in Business Administration.

The first two class meetings were devoted to a description of the system and its inquiry language. Each student was given a user's manual and one two-hour practice session on-line. A monitor was always present to answer questions and assist with technical problems.

For experimental purposes, the class was divided into two sections. Each week, a list of questions covering unfamiliar material was prepared, and one section attempted to answer them using EUREKA while the other group completed the same assignment using the original documents. Class sections alternated from week to week between EUREKA and the documents. In either case, the student was informed of the general subject to be covered and allowed up to two hours of study time. He or she could elect to take the quiz at any time during this period and was then allowed a maximum of one hour in which to complete it.

By proceeding in this way, two important sets of measurements can be obtained. First, we can compare machine assisted searching techniques with the use of conventional materials. Second, we can compare the performance of several motivated users who are all seeking the same information and whose "information need" is known to the investigator.

Twelve quizzes were given during the semester. The first four consisted of short answer-type questions taken from the information retrieval documents. The second set of four quizzes consisted of short answer-type questions taken from the state statutes. The final set consisted of essay questions taken from the state statutes. Overall, the students using the original documents scored approximately 50% better than those using EUREKA.

Throughout the semester, the students using the printed materials spent the preliminary period each week studying and taking notes, while those using EUREKA took considerably less time and used it to reaquaint themselves with the language of the system rather than to study the material. Some sources, e.g., [4], claim that substantial practice is required to develop a facility with an on-line retrieval system. Since each student had two weeks between on-line sessions, they tended to use only the most primitive features (FIND and PRINT statements) of EUREKA. Only one student used EUREKA's comment feature. The DEFINE statement was rarely used, and the MAKE statement was not used at all.

The poor performance of the students using EUREKA can be attributed to two factors in addition to lack of familiarity with the system. First, during the first set of quizzes, the system was quite unstable. Hard failures which occurred during user sessions prevented them from gaining confidence in the system. Thus, they avoided the more powerful features and used EUREKA in a very elementary and time consuming manner. Also, data corruption which occurred on at least two occassions resulted in non-retrieval of relevant documents. Since this type of error did not cause a system crash, it went undetected for an undetermined period of time.

During the second and third sets of quizzes, the system was relatively stable. Unfortunately, the users' opinion of the system was well established by this time. Also, these quizzes were taken from the state statutes. The documents for this data base contain an extensive (500 page) index which was not available on EUREKA. This gave the group using the documents a substantial advantage. For example, one question concerned the

advertisement and sale of birth control devices. The phrase "birth control" does not occur in the text of the statutes but does appear in the index. To retrieve the relevant document using EUREKA, the user must search for some form of the phrase "prevent pregnancy". However, due to their lack of confidence in the system, the group using EUREKA repeatedly searched for forms of the phrase "birth control" often repeating the same search.

The second set of experiments was conducted during the summer semester of 1975. The class consisted of five graduate students and two undergraduates. In order to learn more about training users and to obtain at least a subjective evaluation of the more powerful features of EUREKA, the comparison of machine versus manual searching was temporarily abandoned. The introductory lectures and initial on-line practice sessions were similar to those of the spring semester except that more emphasis was placed on the Macro and Comment features.

The students were given two two-hour sessions per week using EUREKA. Three of these sessions were devoted to short answer quizzes from the spring semester; i.e., exactly the same questions were used in order to make comparisons. Six sessions were devoted to two new essay quizzes which were not used during the spring semester.

During the first short answer quiz, the then inexperienced users approached the system in the same way and performed approximately the same as their confidence-lacking spring semester counterparts. After taking an essay quiz designed to force them to use the more powerful features of EUREKA, they performed substantially better. After another three-session essay, the EUREKA users equalled the performance of the spring semester index-aided document users on the final short answer quiz.

4.2 Feature Evaluation Experiments

Two series of experiments were conducted to evaluate selected features of the EUREKA retrieval system. The evaluation proceedure consisted of giving a set of essay and short answer quizzes to three groups of students - one using the full version of EUREKA, one using a restricted version, and one using the original documents. All questions were taken from the state statutes data base. Since the index to these documents was not available on EUREKA, it was also denied to the document users. The document users still had access to a two-level table of contents which was not available on EUREKA.

The primary emphasis of these experiments was on the relative performance of the group using the restricted version of EUREKA. The document group was retained as an experimental control group. Those features of EUREKA which could be removed without totally handicapping the user were selected for evaluation. The features which were removed are:

1. User personal files
    a. Accessing previous queries
    b. Creating and using macros (personal thesaurus)
    c. Attaching comments to a document or query
2. Full-text searching (ability to search for phrases, to search for the co-occurrence of two or more words in the same sentence or paragraph, and to search user comments)
3. Browse mode (ability to access any portion of a selected document at random).

The subtopics under item 1 can be removed individually or in combinations. Items 1.b. and 1.c. should not affect user performance on short answer quizzes but may be useful on essay quizzes. The other items were expected to have a substantial effect on user and system performance on both types of quizzes.

A cost-benefit analysis of these system features can be developed from the results of these experiments. The benefit of a given feature is defined in terms of the difference in user performance (quiz score) between the group having the feature and the one to which it has been denied. For short answer quizzes, solution time as well as raw score is taken into account in user performance. The development, implementation, and maintenance cost can be estimated by the code and data storage requirements. The cost in terms of system performance is the difference in the system load, defined as the average space-time product per command, between the full system and the restricted system. The space-time product is the amount of core memory required for both code and data multiplied by the CPU time during which it was used.

Table 4.1 gives the storage requirements for the above mentioned features in the current version of EUREKA. Although the User Personal Files feature is a combination of the Macros and Comments and Access to Previous Queries features, the code required is greater than the sum of it parts. This is due to a substantial amount of bookkeeping code which is common to the two parts.

	Disk Data Space (tracks/user)	Code Space (% of total code)	% of Total CPU Time (average user session)
Macros and Comments	4.5	4%	0.01
Previous queries	15	7%	0.5
All personal files	20	20%	0.5
Browse mode	2	3%	15.
Full-text search	0	5%	22.

Table 4.1 Requirements of EUREKA Features

4.2.1 Fall 1975 Experimental Series

The first series of feature evaluation experiments was conducted during the fall semester of 1975. The class consisted of sixteen students - four in Computer Science or related majors, four in Business Administration, and eight in Liberal Arts and Sciences. They were given a users manual, two one-hour lectures covering the system and its inquiry language, one-hour demonstrations in small groups, and two-hour individual practice sessions. After the practice session, they were given a sample short answer-type quiz to complete using EUREKA.

The class was then divided into three groups - one using the full version of EUREKA, one using a restricted version, and one using the original documents. Each group had two two-hour sessions per week. To eliminate inter-group distinction, they rotated every two weeks after taking a three-session essay and a one-session short answer quiz. Two of the essay sessions were devoted to studying a general topic. At the third session, a specific aspect of that topic was assigned and the students were allotted two hours in which to write an essay.

Table 4.2 for the short answer quizzes and Table 4.3 for the essay quizzes present the results of the Fall 1975 series of experiments. Since the short answer quizzes require specific information, user performance is given in terms of points per minute. For essays, however, there is no definite amount of information that is required. For this reason, user performance on the essays is given only in terms of points. The average

Quiz	1		2		3		4		5		6	
System	FULL	FULL	FULL	NOMC	FULL	NOPQ	FULL	NOPF	FULL	NOBR	FULL	NOFT
"THINK" Time (seconds)	28.4	28.6	26.5	22.1	29.9	26.3	23.9	41.2	19.4	21.0	22.5	16.3
Score (points/min)	0.88	0.68	0.94	0.90	1.20	1.53	1.70	0.91	1.13	1.07	2.04	2.90
Space-Time Product per Command	12.1	15.2	48.1	38.0	42.5	49.4	26.6	84.4	47.7	27.3	4.6	1.9
Figure of Merit	.078	.082	.021	.027	.031	.040	.088	.010	.027	.044	0.48	1.71
F test	0.01		0.84		0.32		12.7		3.60		11.9	
Probability	--		--		--		.01		.10		.01	

System - FULL = Full version of EUREKA
NOMC = EUREKA without Macros & Comments
NOPQ = EUREKA without Access to Previous Queries
NOPF = EUREKA without User Personal Files
NOBR = EUREKA without Browse Mode
NOFT = EUREKA without Full-Text Searching

Table 4.2 Fall 1975 Short Answer Quiz Results

Essay	1		2		3		4		5		6	
System	FULL	FULL	FULL	NOMC	FULL	NOPQ	FULL	NOPF	FULL	NOBR	FULL	NOFT
Score (points)	67.6	73.9	74.3	75.2	77.7	79.1	89.8	70.1	75.3	85.5	59.9	71.4
Space-Time Product per Command	31.8	33.3	46.3	13.8	51.5	44.6	14.1	30.7	26.0	20.8	6.3	5.1
Figure of Merit	2.22	2.29	2.69	8.14	1.96	2.54	11.3	5.15	5.13	11.4	27.8	41.0
F test	0.04		8.84		0.46		1.70		2.09		0.38	
Probability	--		.025		--		--		--		--	

System - FULL = Full version of EUREKA
NOMC = EUREKA without Macros & Comments
NOPQ = EUREKA without Access to Previous Queries
NOPF = EUREKA without User Personal Files
NOBR = EUREKA without Browse Mode
NOFT = EUREKA without Full-Text Searching

Table 4.3 Fall 1975 Essay Results

user "THINK" time per command is also shown for short answer quizzes. These times are in the range of mean "THINK" times, 20.0 to 35.3 seconds, found in studies of five time-sharing systems[7]. Since users spend much more time writing during an essay, this parameter does not seem appropriate for essay-type quizzes.

The figure of merit is the average of the ratio of user performance to system load for each user. An analysis of variance was performed for this measure. The values for the F test give the ratio of two estimates of the variance, between groups and within each group, of the figure of merit. A value of F much greater than 1 indicates a larger variance between groups than within each group and therefore a high probability that a difference does exist between the groups. The probability entry in each table was obtained from standard tables for the distribution of F as a function of the sample size of each group, and indicates the probability that the figures of merit are random samplings of the same population.

With the exception of Quiz and Essay #6, the highest statistically significant difference in the figure of merit occurred during Quiz #4 comparing the full system to the system lacking all personal files. User performance is significantly better on the short answer quiz as well as the essay using the full system. Also, system load is drastically increased by the lack of personal files. This may be attributed to the fact that lacking macros and access to previous queries, the user has no alternative to entering long, complicated search requests. This observation is also supported by the substantially longer "THINK" time between queries taken by users of the restricted system on both the short answer quiz and the essay.

A slight improvement in user performance was recorded by users of the restricted system in Essay #5. System load also decreased with the restricted system on this essay and to a greater extent on the short answer quiz. Analysis of user sessions show that users of the restricted system spent approximately twice as much time displaying text as did users of the full system. Since the restricted system did not allow browsing, users were forced to read the entire text of each possibly relevant document. This may have presented them with information which they would have missed if browsing had been available and may therefore have been a factor in their score advantage on the essay. Users of the full system, however, spent less time viewing text and more time performing searches. Searching is naturally more demanding of system resources and accounted for the higher system load.

Essay #2 shows a statistically significant difference in the figure of merit in favor of the system without Macros and Comments. Some small difference is to be expected since full-text searching is required to retrieve user comments. The large difference in this case may be due to lack of familiarity with the feature since, as is shown in Table 4.4, it was not heavily used. Over half the class did not use the Comment feature at all.

For Quiz and Essay #6, the level of indexing was changed because it was felt that inhibiting full-text searching while indexing only to the chapter level would not produce any interesting information. The inverted file was modified to provide pointers to the section level where each section contained approximately 1500 tokens. The actual implementation reformatted the data base, making several smaller new documents (sections) out of the

FEATURE	USEAGE
Macros	Average of less than 1 macro per user per session
Comments	This feature was not used on short answer quizzes. The following statistics are for essays only. Fall 1975: 38% of the users made an average of 10 comments each per two-hour session Spring 1976: 48% of the users made an average of 34 comments each per two-hour session
Access to Previous Queries	40% of the search requests used the results of a previous search
Browse Mode	47% of the time during which users were viewing text, they were browsing
Full-Text Search	Fall 1975: 46% of all searches requested full-text searching Spring 1976: 30% of all searches requested full-text searching

Table 4.4 Useage of EUREKA Features

old documents (chapters). This did not allow term coordination at higher levels; i.e., users could no longer search for co-occurrences of terms at the chapter level.

This was a drastic change for users who had become familiar with the chapter-level indexing since the number of documents increased from 378 to 3176. Therefore, the comparison between the full and restricted systems in this case cannot be considered valid. However, it was noted that users performed significantly better, although somewhat erratic, on this quiz than on previous quizzes. Also, system load decreased drastically. The lower level of indexing reduced the amount of full-text searching required and also provided a new level for term coordination.

User performance on Essay #6 degraded somewhat compared with previous essays. This implementation of section-level indexing tended to fragment concepts since chapters were broken into several documents. This fragmentation did not affect user performance on the short answer quiz since only specific factual information was being sought. However, for the essay, concepts are important and their fragmentation degraded user peformance.

User performance for short answer quizzes throughout the semester are shown in Figure 4.1. The vertical bars indicate the 95% confidence limits in each case. To account for the differences between groups, the average score over all quizzes was calculated for each group. These averages were used as a measure of the native intelligence of each group, and the scores on each quiz were adjusted accordingly. In an attempt to factor out the difficulty of the quiz so that the learning curve could be examined, the scores for each quiz were then normalized to the average document score.

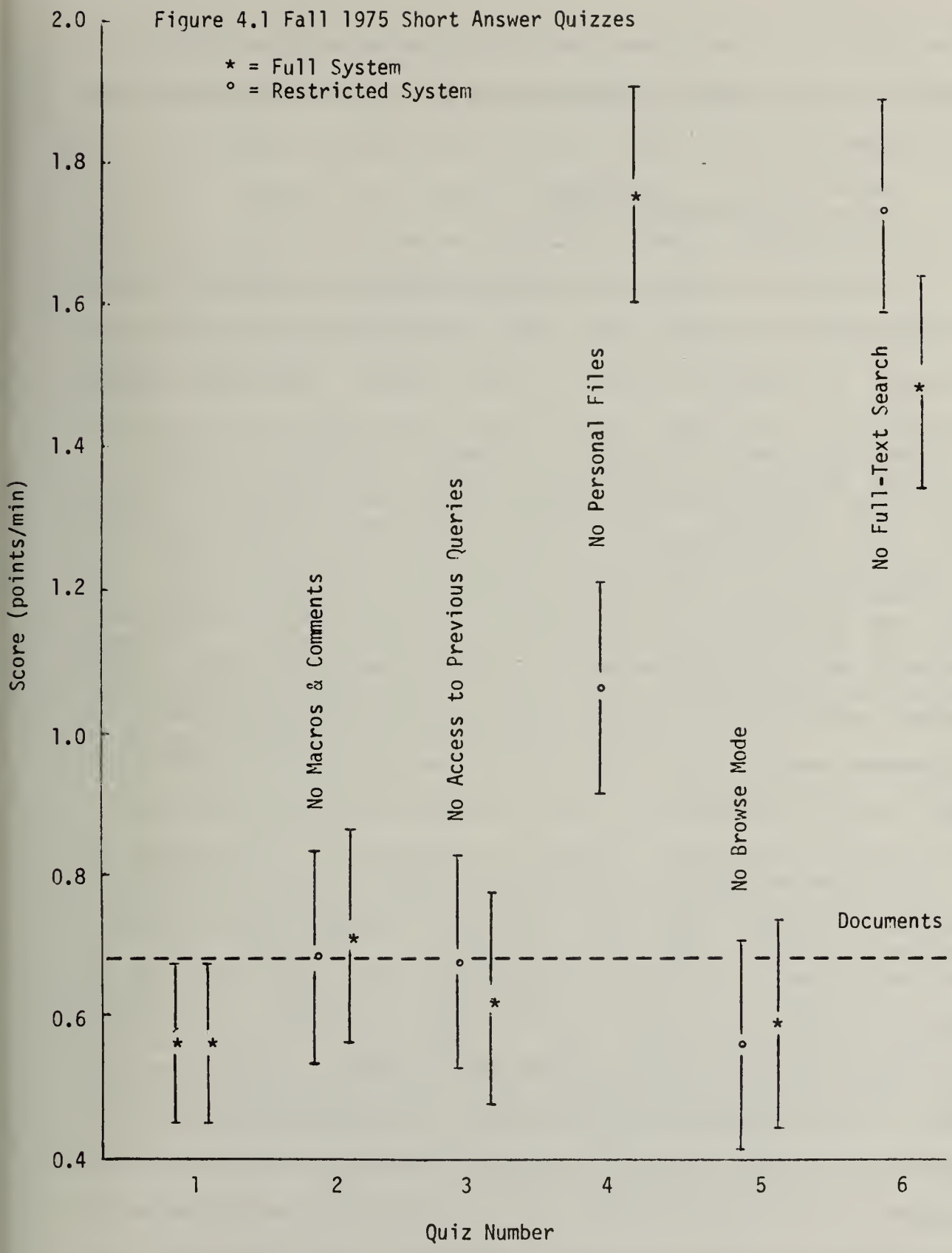


Figure 4.1 Fall 1975 Short Answer Quizzes
* = Full System
° = Restricted System
Score (points/min)
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
No Macros & Comments
No Access to Previous Queries
No Personal Files
No Browse Mode
No Full-Text Search
Documents
1
2
3
4
5
6
Quiz Number

From Figure 4.1, it can be seen that this approximation of the difficulty of the quiz is not valid since Quizzes #4 and #6 were evidently slanted toward the machine.

4.2.2 Spring 1976 Experimental Series

A second set of feature evaluation experiments was conducted during the spring semester of 1976. The class consisted of thirty-one students - fourteen in Computer Science or related majors, nine in Business Administration, and eight in Liberal Arts and Sciences. The introductory lectures, demonstrations, and practice sessions were similar to those in the Fall 1975 experiments. Additionally, this class was given a programming language-type quiz on the EUREKA language after the second lecture. This quiz was designed to force the users to learn the EUREKA language and to obtain an estimate of the native intelligence of each group. This estimate agrees well with the procedure used during the Fall 1975 experiments, the difference between the best group and the worst group being approximately 8%.

The class was again divided into three groups - one using the full version of EUREKA, one using a restricted version, and one using the original documents. For this series of experiments, each group had one one-hour session and one two-hour session per week. To eliminate inter-group distinctions, they rotated every two weeks. During the first week of each two-week period, the one-hour session was devoted to a thirty minute short answer quiz prior to which each student could have up to thirty minutes to refamiliarize himself with the system while the two-hour session was devoted to studying for an essay. During the second week, two hours were allotted for writing an essay and one hour for a short answer quiz.

To investigate the effect of indexing level, all short answer quizzes used the indexing system which was used on Quiz #6 during the Fall 1975 experiments. Because of the concept fragmentation in this system, the chapter-level indexing system was used for all essays. Except for Quiz and Essay #1, the order in which the quizzes were given as well as which restricted system was used with a particular quiz were scrambled from the preceeding semester.

Tables 4.5, 4.6, and 4.7 present the results of these experiments. Analysis of variance of the user performance expressed in points per minute for the short answer quizzes and points for the essays shows no statistical significance between the full version of EUREKA and any restricted version. However, the raw score, which does not include solution time, for Quiz #4 showed a significant difference at the 10% level in favor of the full system over the system lacking access to previous queries.

The results for system performance show a good agreement with the Fall 1975 experiments. As expected, the system lacking full-text searching significantly decreased the system load but also increased the user "THINK" time. User performance on both quizzes and the essay comparing these two systems show a relatively large difference in scores and a large value of F from the analysis of variance. A larger sample size may have shown a statistically significant difference in user performance between these two systems.

Quiz	1		5		2		3		4	
System	FULL	FULL	FULL	NOFT	FULL	NOBR	FULL	NOPF	FULL	NOPQ
"THINK" Time (seconds)	28.6	31.1	21.5	31.5	23.2	21.7	24.5	21.7	20.2	33.9
Score (points/min)	1.16	1.19	1.63	1.32	1.15	1.13	1.72	1.86	2.25	2.32
Space-Time Product per Command	3.4	3.9	18.4	5.4	8.0	4.4	4.6	8.9	6.7	10.1
Figure of Merit	0.46	0.35	0.14	0.30	0.19	0.27	0.41	0.26	0.42	0.52
F test	0.69		8.67		3.72		3.66		0.25	
Probability	—		.01		.10		.10		—	

System - FULL = Full version of EUREKA
NOPQ = EUREKA without Access to Previous Queries
NOPF = EUREKA without User Personal Files
NOBR = EUREKA without Browse Mode
NOFT = EUREKA without Full-Text Searching

Table 4.5 Spring 1976 Short Answer Quiz Results

Essay	1		5		2		3		4		6	
System	FULL	FULL	FULL	NOFT	FULL	NOBR	FULL	NOPF	FULL	NOPQ	FULL	NOMC
Score (points)	65.9	66.3	77.8	73.1	85.1	81.9	83.9	86.3	81.5	77.0	76.5	78.9
Space-Time Product per Command	31.4	31.7	25.7	11.6	32.2	12.6	23.8	76.8	59.8	57.4	49.8	41.8
Figure of Merit	2.23	2.16	5.47	21.1	5.03	10.4	6.05	1.40	2.83	5.90	2.14	2.56
F test	0.02		4.66		3.68		7.32		0.81		0.34	
Probability	--		.05		.10		.025		—		--	

System - FULL = Full version of EUREKA
NOMC = EUREKA without Macros & Comments
NOPQ = EUREKA without Access to Previous Queries
NOPF = EUREKA without User Personal Files
NOBR = EUREKA without Browse Mode
NOFT = EUREKA without Full-Text Searching

Table 4.6 Spring 1976 Essay Results

Quiz	A		B		C		D	
System	FULL	NOFT	FULL	NOBR	FULL	NOPF	FULL	NOPQ
"THINK" Time (seconds)	22.5	28.0	24.1	27.8	19.2	23.7	20.1	30.9
Score (points/min)	1.50	1.16	1.11	1.04	1.96	1.65	1.26	1.12
Space-Time Product per Command	2.8	2.1	13.7	5.9	4.1	12.1	16.9	9.2
Figure of Merit	0.58	0.79	0.13	0.18	0.51	0.15	0.13	0.13
F test	1.57		2.65		18.3		0.01	
Probability	--		--		.001		--	

System - FULL = Full version of EUREKA
NOPQ = EUREKA without Access to Previous Queries
NOPF = EUREKA without User Personal Files
NOBR = EUREKA without Browse Mode
NOFT = EUREKA without Full-Text Searching

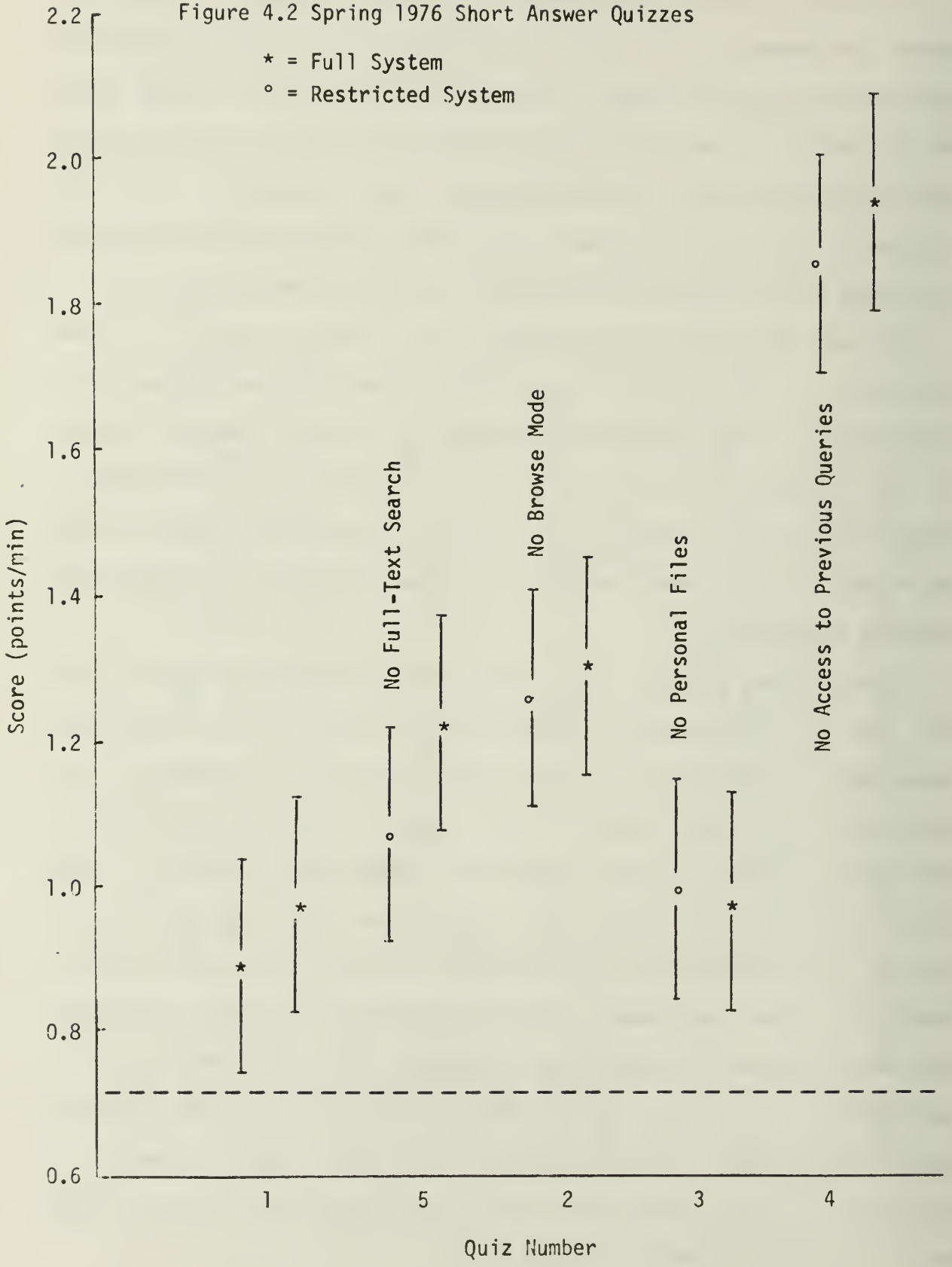
Table 4.7 Spring 1976 Thirty-Minute Short Answer Quiz Results

The system lacking browsing capabilities again showed a decrease in system load compared with the full system. As in the previous experiments, analysis of user sessions showed that users of the restricted system spent 50% to 100% more time viewing text, and thus less time searching, than did users of the full system. The system without user's personal files again displayed a substantial increase in system load while the system lacking only access to previous queries exhibited a relatively small degradation.

The user performance on the one-hour short answer quizzes is shown graphically in Figure 4.2. The scores have been adjusted for the native intelligence of each group and normalized to the overall semester average for the document users. The vertical bars indicate the 95% confidence limits in each case. Comparison with Figure 4.1 shows little change in the performance of the document users while the performance of the EUREKA users increased substantially.

Since the same one-hour short answer quizzes were used during both the Fall and the Spring series of experiments, they can be used to compare chapter-level indexing with section-level indexing by comparing the performance of the group using the full system on each quiz. Table 4.8, a combination of Tables 4.2 and 4.5, shows the comparison between the full version of EUREKA using chapter-level indexing and that using section-level indexing. The user "THINK" time is generally the same using either level of indexing. System performance and user performance are both significantly improved by the use of section-level indexing. The improvement in user performance is statistically significant in four out of the five quizzes while the improvement in system performance and the figure of merit is significant in all five cases. Overall, user performance improved by 50% and system load decreased by a factor of 5.

Figure 4.2 Spring 1976 Short Answer Quizzes



Quiz	1		2		3		4		5	
System	CHAP	SECT	CHAP	SECT	CHAP	SECT	CHAP	SECT	CHAP	SECT
"THINK" Time (seconds)	28.5	29.7	26.5	23.2	29.9	24.5	23.9	20.2	19.4	21.5
Score (points/min)	0.76	1.17	0.94	1.15	1.20	1.72	1.70	2.25	1.13	1.63
Space-Time Product per Command	14.0	5.2	48.1	8.0	42.5	4.6	26.6	6.7	47.7	18.4
Figure of Merit	0.08	0.35	0.02	0.19	0.03	0.41	0.09	0.42	0.03	0.14
F test	16.3		17.2		19.6		11.6		7.78	
Probability	.001		.001		.001		.005		.025	

System - CHAP = Chapter-Level Indexing
SECT = Section-Level Indexing

Table 4.8 Comparison of Chapter- versus Section-Level Indexing

CHAPTER 5 -- ASSESSMENT OF USER ATTITUDES

In addition to user and system performance, another important factor in the evaluation of an information retrieval system is the attitude of users toward the system. Although users may be able to perform adequately with a minimal system when under pressure, they may not willingly use such a system. One method of measuring attitudes is through the use of a semantic differential. A semantic differential consists of a series of bipolar adjective scales on which a subject indicates his reaction to a particular concept. An example is shown in Figure 5.1. One semantic differential exists for each concept to be rated. The subject is instructed to mark one of the seven intervals between each adjective pair indicating the strength of his reaction to the concept.

EUREKA

fast	___:___:___:___:___:___:___	slow
good	___:___:___:___:___:___:___	bad
successful	___:___:___:___:___:___:___	unsuccessful
valuable	___:___:___:___:___:___:___	worthless

Figure 5.1 Example of a Semantic Differential

To reduce the amount of data which must be examined, adjective scales can often be combined into independent groups through factor analysis. Each group then measures a different dimension of a subject's attitude toward a concept.

The adjective scales used for the factor analysis in this evaluation are the same as those used in a 1970 study of SUPARS[8,9], while the concepts which were rated naturally differ. The current study rated fifteen concepts, some of a general nature and some specific to EUREKA. Thirty-two experienced users (four members of the EUREKA staff and twenty-eight students who had participated in the experiments) were given a packet of semantic differentials, one for each concept. The order of the semantic differentials within each packet, the order of the adjective scales within each semantic differential, and the ends of the adjective scales were randomized. The completed semantic differentials were scored and the data was then subjected to factor analysis.

The factor analysis procedure used in this evaluation follows that of Katzer[8]. Each semantic differential was treated as a separate observation resulting in a matrix consisting of 480 observations by 19 variables. The correlation matrix among the variables was first computed. Then the eigenvalues and associated eigenvectors of this matrix were found. To reduce the number of dimensions, only those eigenvalues greater than 1.0 were retained. The remaining dimensions were rotated using Kaiser's varimax procedure[10] to approximate a simple structure.

Each variable was then assigned to the one dimension on which it loaded highest. Acceptable dimensions were required to have at least as many variables assigned to it as the dimensionality of the factor space. For example, the acceptance of a fourth dimension would require each dimension to have at least four variables assigned to it.

The results of the factor analysis are given in Table 5.1. Factor loading is a measure of the correlation between a variable and a dimension. Communality is a measure of the variance of a variable accounted for by the reduced number of dimensions, while factor purity, defined as the square of the highest loading divided by the communality, indicates the proportion of the variance accounted for by the dimension to which the variable is assigned. Variables 11 and 19 loaded highest on a fourth dimension which was discarded because it did not satisfy the requirement for the number of variables assigned to it. Based on factor loading and factor purity values, representative adjective scales, identified by an asterisk in Table 5.1, were then selected from each dimension.

These eight adjective scales were then used for an attitude survey of the students participating in the Spring 1976 series of experiments. The survey was conducted at the end of the semester at which time each student had completed approximately twenty-six contact-hours on EUREKA. Twenty-five of the thirty-one students completed the semantic differential packets.

As in the factor analysis phase, each packet contained fifteen randomized semantic differentials. The completed semantic differentials were scored by assigning integer values from -3 to +3 to the seven-interval adjective scales, positive values indicating a positive reaction. The means and standard deviations were then calculated for each concept by dimension.

VARIABLE #	ADJECTIVE SCALE	FACTOR LOADING I	II	III	COMMUNALITY	FACTOR PURITY
Factor I:	EVALUATION					
1.*	fast-slow	.68	.07	-.21	.52	.89
2.	active-passive	.50	.29	.35	.47	.53
3.*	good-bad	.75	.19	.00	.62	.91
4.	timely-untimely	.58	.26	.03	.40	.83
5.*	successful-unsuccessful	.74	.07	-.10	.60	.92
7.	strong-weak	.53	.29	.36	.59	.48
8.*	valuable-worthless	.81	.11	.04	.67	.97
Factor II:	DESIRABILITY					
9.	hot-cold	.10	.58	.03	.36	.92
10.*	colorful-colorless	.11	.71	.09	.62	.82
12.	kind-cruel	.37	.57	-.29	.56	.59
14.	serious-humorous	.37	-.47	.30	.45	.49
15.*	beautiful-ugly	.28	.72	-.03	.60	.87
17.	frustrating-calming	-.20	-.46	.43	.57	.37
18	friendly-unfriendly	.31	.74	-.14	.69	.78
Factor III:	ENORMITY					
6.	complex-simple	-.06	-.15	.65	.57	.74
13.*	heavy-light	-.08	-.21	.70	.56	.89
16.*	big-small	.06	.12	.75	.59	.95
Not Assigned						
11.	periodic-erratic	.43	-.01	.17	.43	.50
19.	usual-unusual	.02	.05	-.05	.70	.99
	Percent of variance	27.8	12.7	8.7		

Table 5.1 Factor Analysis

Table 5.2 presents the results of the survey. The means are listed together with the standard deviations in parentheses. The concepts are divided into three groups and sorted in descending order of the value of Dimension I within each group. The first group contains general concepts concerning computers and information retrieval systems while the second group contains concepts specific to EUREKA. The only concept in the third group is that of the data base used in the experiments. The first line of each group in Table 5.2 gives the mean and standard deviation for all concepts in that group.

Generally, students indicated a positive reaction to most concepts in all three dimensions. As in the SUPARS study[8], reactions recorded in Dimension I are more pronounced than in the other two dimensions. The more nearly neutral reactions in Dimensions II and III may indicate that these dimensions are not applicable to this type of evaluation.

The concepts in group 2 are of primary importance in this study. It is comforting to note that users indicated a definite positive reaction to EUREKA in general and to most of the specific concepts concerning it. Three features (Browse Mode, Access to Previous Queries, and Full-Text Searching) received a definite positive reaction. Overall, users were neutral toward the Macro and Comment features; however, there was a larger standard deviation for these concepts indicating that some users found them very useful while others considered them worthless. Analysis of user sessions shows that approximately half the students made frequent use of the Comment feature, entering an average of 30 comments per user during the two-hour

	Dimension I Evaluation	Dimension II Desirability	Dimension III Enormity
1. COMPUTERS IN GENERAL	1.61(1.18)	0.74(1.37)	0.34(1.53)
Computer	2.03(1.13)	0.82(1.45)	0.52(1.76)
Computer Search	1.65(1.08)	0.90(1.25)	0.88(1.30)
Terminal	1.62(1.04)	0.46(1.27)	0.06(1.55)
Constructing my Search Logically	1.45(1.16)	0.74(1.31)	-0.02(1.38)
Myself & Computers	1.29(1.43)	0.76(1.56)	0.24(1.61)
2. EUREKA	1.27(1.37)	0.46(1.31)	0.25(1.50)
EUREKA	2.03(1.00)	1.18(1.30)	0.38(1.72)
Browse Mode	1.77(1.18)	0.56(1.26)	0.48(1.63)
Access to Previous Queries	1.59(1.60)	0.60(1.44)	0.12(1.69)
Usefulness of EUREKA to me	1.53(1.28)	0.64(1.39)	0.26(1.27)
Query Language	1.52(1.23)	0.80(1.18)	0.00(1.38)
EUREKA Output	1.42(1.21)	0.16(1.38)	0.38(1.60)
Full-Text Search	1.25(1.18)	0.40(1.12)	1.04(1.18)
Macro Feature	0.37(1.55)	0.06(1.10)	-0.10(1.32)
Comment Feature	-0.03(1.87)	-0.22(1.54)	-0.28(1.61)
3. State Statutes	0.14(1.53)	-0.56(1.49)	0.86(1.45)
COLUMN MEANS	1.31(1.32)	0.49(1.34)	0.32(1.51)

Table 5.2 Semantic Differential Results

study session for each essay (cf. Table 4.4). The Macro feature, however, received only sporadic use throughout the semester.

CHAPTER 6 -- SUMMARY

This research has evaluated several features of a full-text information retrieval system by considering four factors: (1) cost measured by the code and data storage requirements and the percentage of CPU time required during an average user session (cf. Table 4.1), (2) system load measured by the average space-time product per command, (3) user performance measured by quiz score, and (4) user attitude toward the feature. The results of Tables 4.1 through 4.3, 4.5 through 4.7, and 5.2 can be summarized as follows:

Macros and Comments

1. Implementation cost is negligible.
2. System load is increased slightly on essay quizzes because full-text searching is required to access user comments. This feature was not used on short answer quizzes.
3. No significant difference in user performance was noted.
4. User attitude toward this feature was mixed. Approximately half of the students in the Spring 1976 class used the Comment feature heavily and liked it; the other half did not use the feature at all. The Macro feature received little use.

Access to Previous Queries

1. Implementation cost is minor; the code comprises about 7% of the system.
2. System load was decreased somewhat by the presence of this feature.
3. No significant difference in user performance was noted.
4. This feature received a definite positive reaction from users.

User Personal Files

1. Implementation cost is major; the code for this feature is approximately 20% of the system.
2. System load is increased on both types of quizzes by a factor of approximately 2.7 by the absence of this feature.
3. User performance showed a definite improvement with the presence of this feature.
4. This feature was not specifically evaluated in the user attitude survey since it appears to users as a combination of the previous two features.

Browse Mode

1. Implementation cost is negligible.
2. Although the experiments showed a decrease in system load with the absence of this feature, it was shown

that this was due to users having to spend more time viewing text than performing searches.

3. No statistically significant difference in user performance was noted.
4. Users gave this feature the highest rating of all the features of EUREKA.

Full-Text Search

1. Implementation cost is minor.
2. Absence of this feature decreases system load by a factor of approximately 2.3.
3. User performance is consistently better when this feature is available.
4. This feature received a definite positive reaction from users.

Section-Level versus Chapter-Level Indexing

1. There is no difference in the code required; however, the storage required for the section-level inverted file is substantially more than that for chapter-level indexing (20% versus 7% of the full text for the data base used in these experiments).
2. Section-level indexing decreased system load by a factor of approximately 4.3.
3. User performance on short answer quizzes was improved approximately 50% by the use of section-level indexing.

4. Since this concept was somewhat more transparent to users than other concepts, it was not rated on the user attitude survey.

6.1 Extrapolation to Larger Systems

The results presented here were obtained using a system which is too small to be useful in a commercial environment. Most of the results, however, can be scaled to a more realistic system. The implementation costs of the various features, for example, was cited as a percentage of the total code required by the system and should therefore be fairly accurate for other languages on different CPU's. Likewise, the size of the inverted file for different levels of indexing was cited as a percentage of the full text of the documents. Obviously, this will vary somewhat among different data bases depending primarily on the type-token ratio.

Extrapolation of the results for user and system performance, however, is not straightforward. The results presented here were obtained with at most four concurrent users on line. Thus, the CPU was never heavily loaded, and the use of a CPU-intensive operation by one user did not noticably affect response time for other users. In a larger system with more users, the use of those features which increased system load may have an even greater effect on response time and consequently degrade user performance.

6.2 Suggestions for Future Research

The research presented here indirectly evaluated user performance on a word level-indexing system versus that on a section-level indexing system by comparing section-level index with full-text searching to that without full-text searching. Although user performance during the Spring 1976 experiments is consistently better using full-text searching, the results are not statistically significant. This may be due to the increased response time caused by full-text searching. A significant difference might be found by comparing a true word-level indexing system to a section-level indexing system.

Secondly, these experiments evaluted individual restricted systems with the full version of EUREKA. Comparison between restricted systems was not possible since neither the learning curve nor the difficulty of the quiz could be factored out. It would be interesting to conduct a series of experiments to determine the learning curve for users of a system such as EUREKA. Then various user aids designed to shorten the learning process could be evaluated.

Also, the problem of generating thesauri has received considerable attention in the literature. One obvious shortcoming of the EUREKA system is the lack of a thesaurus. The results presented here could be used as the basis for evaluating various methods of automatic or semi-automatic thesaurus generation.

LIST OF REFERENCES

[1] F. W. Lancaster and W. D. Climenson, "Evaluating the Economic Efficiency of a Document Retrieval System," Journal of Documentation, vol. 24, March 1968, pp.16-40.

[2] W. S. Cooper, "Expected Search Length: A Single Measure of Retrieval Effectiveness Based on the Weak Ordering Action of Retrieval Systems," American Documentation, January 1968, pp. 31-40.

[3] J. A. Swets, "Information Retrieval Systems," Science, July 19, 1963, pp. 245-250.

[4] F. W. Lancaster and E. G. Fayen, Information Retrieval On-Line, Los Angeles, Calif.: Melville Publishing Company.

[5] J. H. Carlisle, Interactive Man-Machine Communication, New Haven, Conn.: Yale University.

[6] J. K. Morgan, "Description of an Experimental On-Line Minicomputer-Based Information Retrieval System," M. S. Thesis, University of Illinois Department of Computer Science Report Number 76-779, February 1976.

[9] H. Sackman, Man-Computer Problem Solving, Auerbach Publishers Inc., 1970.

[8] J. Katzer, "The Development of a Semantic Differential to Assess Users' Attitudes toward an On-Line Interactive Reference Retrieval System," Journal of the American Society for Information Science, March-April 1972, pp. 122-127.

[9] E. Frierson and P. Atherton, "Survey of Attitudes toward SUPARS," Proceedings of the American Society for Information Science, vol. 8, 1971, pp. 65-69.

[10] H. H. Harman, Modern Factor Analysis, The University of Chicago Press, 1960.

VITA

James Richard Rinewalt was born on May 10, 1946 in Waco, Texas. He received a B.S. degree in Electrical Engineering with High Honors in 1969 and a M.S. degree in Electrical Engineering in 1971, both from the University of Texas at Arlington. While an undergraduate, he was a Cooperative Education Student with LTV Aerospace Corp. in Dallas, Texas.

Mr. Rinewalt was employed by LTV Aerospace Corp. as an Associate Electronics Systems Engineer during 1969 and 1970. In 1971, he received the LTV Graduate Scholarship. He also received the LTV Doctoral Fellowship award in 1972 and 1973. From 1971 through 1976, he was a Research Assistant in the Department of Computer Science at the University of Illinois.

Mr. Rinewalt is a member of Eta Kappa Nu and Tau Beta Pi.

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-76-823	2.	3. Recipient's Accession No.

4. Title and Subtitle

EVALUATION OF SELECTED FEATURES OF THE EUREKA FULL-TEXT INFORMATION RETRIEVAL SYSTEM

5. Report Date: September 1976

6.

7. Author(s): James Richard Rinewalt

8. Performing Organization Rept. No. UIUCDCS-R-76-823

9. Performing Organization Name and Address

University of Illinois at Urbana-Champaign
Department of Computer Science
Urbana, Illinois 61801

10. Project/Task/Work Unit No.

11. Contract/Grant No. US NSF DCR73-07980 A02

12. Sponsoring Organization Name and Address

National Science Foundation
Washington, D. C.

13. Type of Report & Period Covered: Doctoral - 1976

14.

15. Supplementary Notes

16. Abstracts

Evaluations of on-line information retrieval systems have been largely dependent upon monitoring of users' searches and the cooperation of users in interviews and questionnaires. Since users have a variety of information needs and levels of experience, the evaluation process has been difficult.

This paper describes and presents the results of a series of experiments designed to evaluate various features of an information retrieval system in a controlled environment. Features are evaluated on the basis of their development and implementation cost, their effect on system performance as well as user performance, and the attitude of users toward the feature.

17. Key Words and Document Analysis. 17a. Descriptors

Feature Evaluation
Full-Text Searching
Information Retrieval
Inverted Files
Man-Machine Interaction
User Behavior

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Release Unlimited	UNCLASSIFIED	60
	20. Security Class (This Page) UNCLASSIFIED	22. Price